Double Ramp Loss Based Reject Option Classifier

Naresh Manwani[1](✉), Kalpit Desai[2],
Sanand Sasidharan[1], and Ramasubramanian Sundararajan[3]

[1] Data Mining Laboratory, GE Global Research, JFWTC, Whitefield,
Bangalore, India
{Naresh.Manwani,Sanand.Sasidharan}@ge.com

[2] Bidgely Technologies Pvt Ltd., Bangalore, India
kvdesai@gmail.com

[3] Sabre Airline Solutions, Bangalore, India
gs.ramsu@gmail.com

Abstract. The performance of a reject option classifiers is quantified using $0-d-1$ loss where $d \in (0, .5)$ is the loss for rejection. In this paper, we propose *double ramp loss* function which gives a continuous upper bound for $(0-d-1)$ loss. Our approach is based on minimizing regularized risk under the double ramp loss using *difference of convex programming*. We show the effectiveness of our approach through experiments on synthetic and benchmark datasets. Our approach performs better than the state of the art reject option classification approaches.

1 Introduction

The primary focus of classification problems has been on algorithms that return a prediction on every example. However, in many real life situations, it may be prudent to *reject* an example rather than run the risk of a costly potential misclassification. Consider, for instance, a physician who has to return a diagnosis for a patient based on the observed symptoms and a preliminary examination. If the symptoms are either ambiguous, or rare enough to be unexplainable without further investigation, then the physician might choose not to risk misdiagnosing the patient. He might instead ask for further medical tests to be performed, or refer the case to an appropriate specialist. The principal response in these cases is to "reject" the example. This paper focuses on learning a classifier with a reject option. From a geometric standpoint, we can view the classifier as being possessed of a decision surface as well as a rejection surface. The rejection region impacts the proportion of examples that are likely to be rejected, as well as the proportion of predicted examples that are likely to be correctly classified. A well-optimized classifier with a reject option is the one which minimizes the rejection rate as well as the mis-classification rate on the predicted examples.

Let $\mathbf{x} \in \mathbb{R}^p$ is the feature vector and $y \in \{-1, +1\}$ is the class label. Let $\mathcal{D}(\mathbf{x}, y)$ be the joint distribution of $\mathbf{x}$ and y. A typical *reject option classifier* is defined using a bandwidth parameter (ρ) and a separating surface $(f(\mathbf{x}) = 0)$.

T. Cao et al. (Eds.): PAKDD 2015, Part I, LNAI 9077, pp. 151–163, 2015.
DOI: 10.1007/978-3-319-18038-0_12

ρ is the parameter which determines the rejection region. Then a reject option classifier $h(f(\mathbf{x}), \rho)$ is formed as:

$$h(f(\mathbf{x}), \rho) = 1.\mathbb{I}_{\{f(\mathbf{x})>\rho\}} + 0.\mathbb{I}_{\{|f(\mathbf{x})|\leq\rho\}} - 1.\mathbb{I}_{\{f(\mathbf{x})<-\rho\}} \tag{1}$$

where $\mathbb{I}_{\{A\}}$ is an indicator function which takes value 1 if predicate 'A' is true, else 0. The reject option classifier can be viewed as two parallel surfaces with the rejection area in between. The goal is to determine $f(\mathbf{x})$ as well as ρ simultaneously. The performance of this classifier is evaluated using L_{0-d-1} [8,12] which is

$$L_{0-d-1}(f(\mathbf{x}), y, \rho) = 1.\mathbb{I}_{\{yf(\mathbf{x})<-\rho\}} + d.\mathbb{I}_{\{|f(\mathbf{x})|\leq\rho\}} + 0.\mathbb{I}_{\{yf(\mathbf{x})\geq-\rho\}} \tag{2}$$

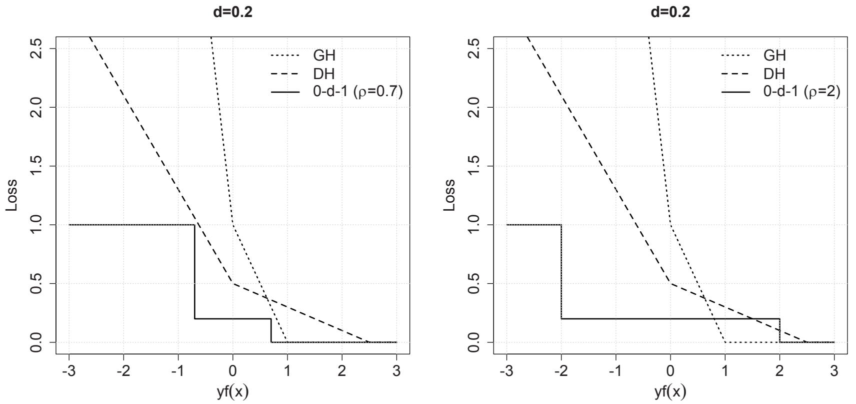
In the above loss, d is the cost of rejection. If $d = 0$, then we will always reject. When $d > .5$, then we will never reject (because expected loss of random labeling is 0.5). Thus, we always take $d \in (0, .5)$.

To learn a reject option classifier, the expectation of $L_{0-d-1}(.,.,.)$ with respect to $\mathcal{D}(\mathbf{x}, y)$ (*risk*) is minimized. Since $\mathcal{D}(\mathbf{x}, y)$ is fixed but unknown, the empirical risk minimization principle is used. The risk under L_{0-d-1} is minimized by *generalized Bayes discriminant* [4,8]. $h(f(\mathbf{x}), \rho)$ (Eq. (1)) is shown to be infinite sample consistent with respect to the generalized Bayes classifier [13].

Table 1. Convex surrogates for L_{0-d-1}

Loss Function	**Definition**
Generalized Hinge	$L_{\text{GH}}(f(\mathbf{x}), y) = \begin{cases} 1 - \frac{1-d}{d} yf(\mathbf{x}), & \text{if } yf(\mathbf{x}) < 0 \\ 1 - yf(\mathbf{x}), & \text{if } 0 \leq yf(\mathbf{x}) < 1 \\ 0, & \text{otherwise} \end{cases}$
Double Hinge	$L_{\text{DH}}(f(\mathbf{x}), y) = \max[-y(1-d)f(\mathbf{x}) + H(d), -ydf(\mathbf{x}) + H(d), 0]$ where $H(d) = -d\log(d) - (1-d)\log(1-d)$

Since minimizing the risk under L_{0-d-1} is computationally cumbersome, convex surrogates for L_{0-d-1} have been proposed. *Generalized hinge loss* L_{GH} (see Table 1) is a convex surrogate for L_{0-d-1} [3,12]. It is shown that a minimizer of risk under L_{GH} is consistent to the generalized Bayes classifier [3]. *Double hinge loss* L_{DH} (see Table 1) is another convex surrogate for L_{0-d-1} [7]. Minimizer of the risk under L_{DH} is shown to be *strongly universally consistent* to the generalized Bayes classifier [7]. We observe that these convex loss functions have some limitations. For example, L_{GH} is a convex upper bound to L_{0-d-1} provided $\rho < 1 - d$ and L_{DH} forms an upper bound to L_{0-d-1} provided $\rho \in (\frac{1-H(d)}{1-d}, \frac{H(d)-d}{d})$ (see Fig. 1). Also, both L_{GH} and L_{DH} increase linearly in the rejection region instead of remaining constant. These convex losses can become unbounded for misclassified examples with the scaling of parameters of f. Moreover, limited experimental results are shown to validate the practical significance of these losses [3,7,12]. A non-convex formulation for learning reject



(a)

(b)

Fig. 1. L_{GH} and L_{DH} for $d = 0.2$. (a) For $\rho = 0.7$, both the losses upper bound the L_{0-d-1}. For $\rho = 2$, both the losses fail to upper bound L_{0-d-1}. L_{GH} and L_{DH} both increase linearly even in the rejection region than being flat.

option classifier is proposed in [5]. However, theoretical guarantees for the approach proposed in [5] are not known. While learning a reject option classifier, one has to deal with the overlapping class regions and outliers. SVM and other convex loss based approaches are less robust to label noise and outliers in the data [10]. It is shown that ramp loss based approach is more robust to noise [6].

Motivated by this, we propose *double ramp loss* (L_{DR}) which incorporates a different loss value for rejection. L_{DR} forms a continuous nonconvex upper bound for L_{0-d-1} and overcomes many of the issues of convex surrogates of L_{0-d-1}. To learn a reject option classifier, we minimize the regularized risk under L_{DR} which becomes an instance of difference of convex (DC) functions. To minimize it, we use DC programming approach [1]. The proposed method has following advantages: (1) the proposed loss L_{DR} gives a tighter upper bound to the L_{0-d-1}, (2) L_{DR} requires no constraint on ρ unlike L_{GH} and L_{DH}, (3) our approach can be easily kernelized for dealing with nonlinear problems.

The rest of the paper is organized as follows. In Section 2 we define the *double ramp loss* (L_{DR}). Then we discuss its properties and the proposed formulation based on L_{DR}. In Section 3 we derive the (L_{DR}) based reject option classifier learning algorithm. We present experimental results in Section 4. We conclude the paper with the discussion in Section 5.

2 Proposed Approach

Our approach for learning classifier with reject option is based on minimizing regularized risk under L_{DR} (double ramp loss).

2.1 Double Ramp Loss

Double ramp loss is defined as a sum of two ramp loss functions as follows:

$$\begin{aligned} L_{\mathrm{DR}}(f(\mathbf{x}), y, \rho) = &\frac{d}{\mu}\Big[\big[\mu - yf(\mathbf{x}) + \rho\big]_+ - \big[-\mu^2 - yf(\mathbf{x}) + \rho\big]_+\Big] \\ &+ \frac{(1-d)}{\mu}\Big[\big[\mu - yf(\mathbf{x}) - \rho\big]_+ - \big[-\mu^2 - yf(\mathbf{x}) - \rho\big]_+\Big] \quad (3) \end{aligned}$$

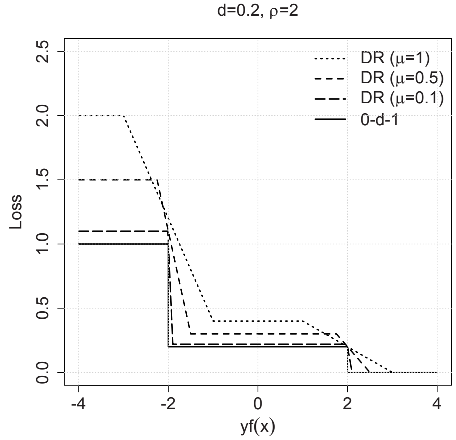


Fig. 2. L_{DR} and L_{0-d-1} : $\forall \mu \geq 0, \rho \geq 0$, L_{DR} is an upper bound for L_{0-d-1}

where $[a]_+ = \max(0, a)$. $\mu \in (0, 1]$ defines the slope of ramps in the loss[1]. Parameter ρ defines the width of the rejection region. Fig. 2 shows L_{DR} for $d = 0.2, \rho = 2$ for different μ.

Theorem 1. *(i) $L_{DR} \geq L_{0-d-1}, \forall \mu > 0, \rho \geq 0$. (ii) $\lim_{\mu \to 0} L_{DR}(f(\mathbf{x}), \rho, y) = L_{0-d-1}(f(\mathbf{x}), \rho, y)$. (iii) In the rejection region, $yf(\mathbf{x}) \in (\rho - \mu^2, -\rho + \mu)$, $L_{DR}(f(\mathbf{x}), y, \rho) = d(1 + \mu)$, a const. (iv) $L_{DR} \leq (1 + \mu), \forall \rho \geq 0, d \geq 0$. (v) When $\rho = 0$, L_{DR} is same as μ-ramp loss ([11]). (vi) L_{DR} is a non-convex function of $(yf(\mathbf{x}), \rho)$.*

The proof of Theorem 1 is omitted due to the space constraints. We see that L_{DR} does not put any restriction on ρ for it to be an upper bound of L_{0-d-1}.

2.2 Risk Formulation Using L_{DR}

Let $\mathcal{S} = \{(\mathbf{x}_n, y_n),\ n = 1 \ldots N\}$ be the training dataset, where $\mathbf{x}_n \in \mathbb{R}^p$, $y_n \in \{-1, +1\}$, $\forall n$. As discussed, we minimize regularized risk under L_{DR} to find

[1] While L_{DR} is parametrized by μ and d as well, we omit them for the sake of notational consistency.

a reject option classifier. In this paper, we use l_2 regularization. Let $\Theta = [\mathbf{w}^T \quad b \quad \rho]^T$. Thus, for $f(\mathbf{x}) = (\mathbf{w}^T\phi(\mathbf{x}) + b)$, regularized risk under double ramp loss is

$$\begin{aligned} R(\Theta) &= \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{\mu}\sum_{n=1}^{N}\Big\{d\big[\mu - y_nf(\mathbf{x}_n)+\rho\big]_+ - d\big[-\mu^2 - y_nf(\mathbf{x}_n)+\rho\big]_+ \\ &\quad +(1-d)\big[\mu - y_nf(\mathbf{x}_n)-\rho\big]_+ - (1-d)\big[-\mu^2 - y_nf(\mathbf{x}_n)-\rho\big]_+\Big\} \\ &= \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{\mu}\sum_{n=1}^{N}\Big\{d\big[\mu - y_nf(\mathbf{x}_n)+\rho\big]_+ + (1-d)\big[\mu - y_nf(\mathbf{x}_n)-\rho\big]_+ \\ &\quad -d\big[-\mu^2 - y_nf(\mathbf{x}_n)+\rho\big]_+ - (1-d)\big[-\mu^2 - y_nf(\mathbf{x}_n)-\rho\big]_+\Big\} \end{aligned}$$

where C is regularization parameter. While minimizing $R(\Theta)$, no non-negativity condition on ρ is required due to the following lemma.

Lemma 1. *At the minimum of $R(\Theta)$, ρ must be non-negative.*

Proof. Let $\Theta' = (\mathbf{w}', b', \rho')$ minimizes $R(\Theta)$, where $\rho' < 0$. Thus $-\rho' > 0$. Consider $\Theta'' = (\mathbf{w}', b', -\rho')$ as another point.

$$\begin{aligned} R(\Theta') - R(\Theta'') &= \frac{C(1-2d)}{\mu}\sum_{n=1}^{N}\Big\{-\big[\mu - y_nf(\mathbf{x}_n)+\rho'\big]_+ + \big[-\mu^2 - y_nf(\mathbf{x}_n)+\rho'\big]_+ \\ &\quad +\big[\mu - y_nf(\mathbf{x}_n)-\rho'\big]_+ - \big[-\mu^2 - y_nf(\mathbf{x}_n)-\rho'\big]_+\Big\} \\ &= C(1-2d)\sum_{n=1}^{N}\Big\{L_{ramp}(y_nf(\mathbf{x}_n)+\rho') - L_{ramp}(y_nf(\mathbf{x}_n)-\rho')\Big\} \end{aligned}$$

where $L_{ramp}(t) = \frac{1}{\mu}([\mu - t]_+ - [-\mu^2 - t]_+)$ is a monotonically non-increasing function of t [11]. Since $\rho' < 0$, thus, $y_nf(\mathbf{x}_n) + \rho' < y_nf(\mathbf{x}_n) - \rho'$, $\forall n$. This implies $L_{ramp}(y_nf(\mathbf{x}_n)+\rho') \geq L_{ramp}(y_nf(\mathbf{x}_n)-\rho')$, $\forall n$. Also $(1-2d) \geq 0$, since $0 \leq d \leq 0.5$. Thus $R(\Theta') - R(\Theta'') \geq 0$, which contradicts that Θ' minimizes $R(\Theta)$. Thus, at the minimum of $R(\Theta)$, ρ must be non-negative.

3 Solution Methodology

$R(\Theta)$ (Eq. (4)) is a nonconvex function of Θ. However, $R(\Theta)$ can be written as $R(\Theta) = R_1(\Theta) - R_2(\Theta)$, where $R_1(\Theta)$ and $R_2(\Theta)$ are convex functions of Θ.

$$R_1(\Theta) = \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{\mu}\sum_{n=1}^{N}\Big[d\big[\mu - y_nf(\mathbf{x}_n)+\rho\big]_+ + (1-d)\big[\mu - y_nf(\mathbf{x}_n)-\rho\big]_+\Big]$$

$$R_2(\Theta) = \frac{C}{\mu}\sum_{n=1}^{N}\Big[d\big[-\mu^2 - y_nf(\mathbf{x}_n)+\rho\big]_+ + (1-d)\big[-\mu^2 - y_nf(\mathbf{x}_n)-\rho\big]_+\Big]$$

In this case, DC programming guarantees to find a local optima of $R(\Theta)$ [1]. In the simplified DC algorithm [1], an upper bound of $R(\Theta)$ is found using the convexity property of $R_2(\Theta)$ as follows.

$$R(\Theta) \leq R_1(\Theta) - R_2(\Theta^{(l)}) - (\Theta - \Theta^{(l)})^T \nabla R_2(\Theta^{(l)}) =: ub(\Theta, \Theta^{(l)}) \tag{4}$$

where $\Theta^{(l)}$ is the parameter vector after $(l)^{th}$ iteration, $\nabla R_2(\Theta^{(l)})$ is a subgradient of R_2 at $\Theta^{(l)}$. $\Theta^{(l+1)}$ is found by minimizing $ub(\Theta, \Theta^{(l)})$. Thus, $R(\Theta^{(l+1)}) \leq ub(\Theta^{(l+1)}, \Theta^{(l)}) \leq ub(\Theta^{(l)}, \Theta^{(l)}) = R(\Theta^{(l)})$. Which means, in every iteration, the DC program reduces the value of $R(\Theta)$.

3.1 Learning Reject Option Classifier Using DC Programming

In this section, we will derive a DC algorithm for minimizing $R(\Theta)$. We initialize with $\Theta = \Theta^{(0)}$. Given $\Theta^{(l)}$, we find $\Theta^{(l+1)}$ as

$$\Theta^{(l+1)} \in \arg\min_{\Theta} \; ub(\Theta, \Theta^{(l)}) = \arg\min_{\Theta} \; R_1(\Theta) - \Theta^T \nabla R_2(\Theta^{(l)}) \tag{5}$$

where $\nabla R_2(\Theta^{(l)})$ is the subgradient of $R_2(\Theta)$ at $\Theta^{(l)}$. We choose $\nabla R_2(\Theta^{(l)})$ as:

$$\nabla R_2(\Theta^{(l)}) = \sum_{n=1}^{N} \beta_n'^{(l)} [-y_n \phi(\mathbf{x}_n)^T \quad -y_n \quad 1]^T + \sum_{n=1}^{N} \beta_n''^{(l)} [-y_n \phi(\mathbf{x}_n)^T \quad -y_n \quad -1]^T$$

where

$$\begin{cases} \beta_n'^{(l)} = \frac{Cd}{\mu} \mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l)} + b^{(l)}) - \rho^{(l)} < -\mu^2\}} \\ \beta_n''^{(l)} = \frac{C(1-d)}{\mu} \mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l)} + b^{(l)}) + \rho^{(l)} < -\mu^2\}} \end{cases} \tag{6}$$

For $f(\mathbf{x}) = (\mathbf{w}^T \phi(\mathbf{x}) + b)$, we rewrite the upper bound minimization problem described in Eq. (5) as follows,

$$\begin{aligned} P^{(l+1)} &= \min_{\Theta} \; R_1(\Theta) - \Theta^T \nabla R_2(\Theta^{(l)}) \\ &= \min_{\mathbf{w},b,\rho} \; \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{\mu} \sum_{n=1}^{N} \Big[d\big[\mu - y_n f(\mathbf{x}_n) + \rho\big]_+ + (1-d)\big[\mu - y_n f(\mathbf{x}_n) - \rho\big]_+ \Big] \\ &\qquad + \sum_{n=1}^{N} \beta_n'^{(l)} [y_n f(\mathbf{x}_n) - \rho] + \sum_{n=1}^{N} \beta_n''^{(l)} [y_n f(\mathbf{x}_n) + \rho] \end{aligned}$$

We rewrite $P^{(l+1)}$ as

$$\begin{aligned} P^{(l+1)} = &\min_{\mathbf{w},b,\boldsymbol{\xi}',\boldsymbol{\xi}'',\rho} \; \frac{1}{2}||\mathbf{w}||^2 + \frac{C}{\mu} \sum_{n=1}^{N} \left[d\xi_n' + (1-d)\xi_n'' \right] + \sum_{n=1}^{N} \beta_n'^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \rho] \\ &\qquad + \sum_{n=1}^{N} \beta_n''^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] \\ &s.t. \quad y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq \rho + \mu - \xi_n', \quad \xi_n' \geq 0, \quad n = 1 \ldots N \\ &\qquad\; y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq -\rho + \mu - \xi_n'', \quad \xi_n'' \geq 0 \quad n = 1 \ldots N \end{aligned}$$

where $\boldsymbol{\xi}' = [\xi'_1 \quad \xi'_2 \ldots \xi'_N]^T$ and $\boldsymbol{\xi}'' = [\xi''_1 \quad \xi''_2 \ldots \xi''_N]^T$. The dual optimization problem $D^{(l+1)}$ of $P^{(l+1)}$ is as follows.

$$D^{(l+1)} = \min_{\boldsymbol{\gamma}',\boldsymbol{\gamma}''} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m (\gamma'_n + \gamma''_n)(\gamma'_m + \gamma''_m) k(\mathbf{x}_n, \mathbf{x}_m) - \mu \sum_{n=1}^{N} (\gamma'_n + \gamma''_n)$$
$$s.t. \begin{cases} -\beta_n'^{(l)} \leq \gamma'_n \leq \frac{Cd}{\mu} - \beta_n'^{(l)} & n = 1 \ldots N \\ -\beta_n''^{(l)} \leq \gamma''_n \leq \frac{C(1-d)}{\mu} - \beta_n''^{(l)} & n = 1 \ldots N \\ \sum_{n=1}^{N} y_n(\gamma'_n + \gamma''_n) = 0 \quad \sum_{n=1}^{N} (\gamma'_n - \gamma''_n) = 0 \end{cases}$$

where $\boldsymbol{\gamma}' = [\gamma'_1 \quad \gamma'_2 \ldots\ldots \gamma'_n]^T$ and $\boldsymbol{\gamma}'' = [\gamma''_1 \quad \gamma''_2 \ldots\ldots \gamma''_n]^T$ are dual variables. At the optimality of $P^{(l+1)}$, $\mathbf{w}$ can be found as $\mathbf{w} = \sum_{n=1}^{N} y_n(\gamma'_n + \gamma''_n)\phi(\mathbf{x}_n)$.

Since $P^{(l+1)}$ has quadratic objective and linear constraints, it holds strong duality with $D^{(l+1)}$. Solving $D^{(l+1)}$ is more useful as it can be easily kernelized for non-linear problems. Behavior of γ'_n and γ''_n under different cases is as follows.

$$\begin{cases} y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - \mu > \rho & \Rightarrow \gamma'_n = -\beta_n'^{(l)};\ \ \gamma''_n = -\beta_n''^{(l)} \\ y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - \mu = \rho & \Rightarrow \gamma'_n \in \big(-\beta_n'^{(l)}, \frac{Cd}{\mu} - \beta_n'^{(l)}\big);\ \ \gamma''_n = -\beta_n''^{(l)} \\ y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - \mu \in (-\rho, \rho) & \Rightarrow \gamma'_n = \frac{Cd}{\mu} - \beta_n'^{(l)};\ \ \gamma''_n = -\beta_n''^{(l)} \\ y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - \mu = -\rho & \Rightarrow \gamma'_n = \frac{Cd}{\mu} - \beta_n'^{(l)};\ \ \gamma''_n \in \big(-\beta_n''^{(l)}, \frac{C(1-d)}{\mu} - \beta_n''^{(l)}\big) \\ y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - \mu < -\rho & \Rightarrow \gamma'_n = \frac{Cd}{\mu} - \beta_n'^{(l)};\ \ \gamma''_n = \frac{C(1-d)}{\mu} - \beta_n''^{(l)} \end{cases}$$

3.2 Finding $b^{(l+1)}$ and $\rho^{(l+1)}$

To find $b^{(l+1)}$ and $\rho^{(l+1)}$, we consider $\mathbf{x}_n \in \mathrm{SV}'^{(l+1)} \cup \mathrm{SV}''^{(l+1)}$, where

$$\mathrm{SV}'^{(l+1)} = \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T\mathbf{w}^{(l+1)} + b^{(l+1)}) = \rho^{(l+1)} + \mu\}$$
$$\mathrm{SV}''^{(l+1)} = \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T\mathbf{w}^{(l+1)} + b^{(l+1)}) = -\rho^{(l+1)} + \mu\}$$

We already saw that

1. If $\mathbf{x}_n \in \mathrm{SV}'^{(l+1)}$, then $\gamma_n'^{(l+1)} \in \big(-\beta_n'^{(l)}, \frac{Cd}{\mu} - \beta'_n(l)\big)$ and $\gamma_n''^{(l+1)} = -\beta_n''^{(l)}$
2. If $\mathbf{x}_n \in \mathrm{SV}''^{(l+1)}$, then $\gamma_n'^{(l+1)} = \frac{Cd}{\mu} - \beta_n'^{(l)}$ and $\gamma_n''^{(l+1)} \in \big(-\beta_n''^{(l)}, \frac{C(1-d)}{\mu} - \beta_n''^{(l)}\big)$

We solve the system of linear equations corresponding to sets $\mathrm{SV}'^{(l+1)}$ and $\mathrm{SV}''^{(l+1)}$ for identifying $b^{(l+1)}$ and $\rho^{(l+1)}$.

3.3 Summary of the Algorithm

We fix $d \in [0, .5]$, $\mu \in (0, 1]$ and C and initialize the parameter vector Θ as $\Theta^{(0)}$. In any iteration (l), we find $\beta_n'^{(l)}, \beta_n''^{(l)},\ n = 1 \ldots N$ (see Eq. (6)). We solve

$D^{(l+1)}$ to find $\boldsymbol{\gamma}'^{(l+1)}, \boldsymbol{\gamma}''^{(l+1)}$. $\mathbf{w}^{(l+1)}$ is found as $\mathbf{w}^{(l+1)} = \sum_{n=1}^{N} y_n(\gamma_n'^{(l+1)} + \gamma_n''^{(l+1)})\phi(\mathbf{x}_n)$. We find $b^{(l+1)}$ and $\rho^{(l+1)}$ as described in Section 3.2. Thus, we have found $\Theta^{(l+1)}$. Using $\Theta^{(l+1)}$, we now find $\beta_n'^{(l+1)}, \beta_n''^{(l+1)},\ n = 1 \ldots N$. We repeat the above two steps until the parameter vector Θ changes significantly. More formal description of our algorithm is provided in Algorithm 1.

Algorithm 1. Learning Reject Option Classifier by Minimizing $R(\Theta)$

```
Input : $d \in [0, .5],\ \mu \in (0, 1],\ C > 0,\ \mathcal{S}$
Output : $\mathbf{w}^*, b^*, \rho^*$
Initialize $\mathbf{w}^{(0)}, b^{(0)}, \rho^{(0)},\ l = 0$
repeat
    Compute $\beta_n'^{(l)} = \frac{Cd}{\mu}\mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T\mathbf{w}^{(l)}+b^{(l)})-\rho^{(l)}<-\mu^2\}}$
    $\beta_n''^{(l)} = \frac{C(1-d)}{\mu}\mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T\mathbf{w}^{(l)}+b^{(l)})+\rho^{(l)}<-\mu^2\}}$
    Find $\boldsymbol{\gamma}'^{(l+1)}, \boldsymbol{\gamma}''^{(l+1)}$ by solving $D^{(l+1)}$ described in Eq. (7)
    Find $\mathbf{w}^{(l+1)} = \sum_{n=1}^{N} y_n(\gamma_n'^{(l+1)} + \gamma_n''^{(l+1)})\phi(\mathbf{x}_n)$
    Find $b^{(l+1)}$ and $\rho^{(l+1)}$ by solving the system of linear equations corresponding to
    sets $\mathrm{SV}_1^{(l+1)}$ and $\mathrm{SV}_2^{(l+1)}$, where
        $\mathrm{SV}'^{(l+1)} = \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T\mathbf{w}^{(l+1)} + b^{(l+1)}) = \rho^{(l+1)} + \mu\}$
        $\mathrm{SV}''^{(l+1)} = \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T\mathbf{w}^{(l+1)} + b^{(l+1)}) = -\rho^{(l+1)} + \mu\}$
until convergence of $\Theta^{(l)}$
```

3.4 γ' and γ'' at the Convergence of Algorithm 1

At the convergence of Algorithm 1, let $\gamma_n'^*, \gamma_n''^*,\ n = 1 \ldots N$ become the values of the dual variables. The behavior of $\gamma_n'^*$ and $\gamma_n''^*$ is described in Table 2. For any $\mathbf{x}_n$, only one of $\gamma_n'^*$ and $\gamma_n''^*$ can be nonzero. We observe that parameters $\mathbf{w}, b$ and ρ are determined by the points whose margin $(yf(\mathbf{x}))$ is in the range $[\rho-\mu^2, \rho+\mu] \cup [-\rho-\mu^2, -\rho+\mu]$. We call these points as *support vectors.* We also see that $\mathbf{x}_n$ for which $y_n f(\mathbf{x}_n) \in (\rho + \mu, \infty) \cup (-\rho + \mu, \rho - \mu^2) \cup (-\infty, -\rho - \mu^2)$, both $\gamma_n'^*, \gamma_n''^* = 0$. Thus, points which are correctly classified with margin at least $(\rho + \mu)$, points falling close to the decision boundary with margin in the interval $(-\rho + \mu, \rho - \mu^2)$ and points misclassified with a high negative margin (less than $-\rho - \mu^2$), are ignored in the final classifier. Thus, our approach not only rejects points falling in the overlapping region of classes, it also ignores potential outliers. We illustrate these insights through experiments on a synthetic dataset as shown in Fig. 3. 400 points are uniformly sampled from the square region $[0\ \ 1] \times [0\ \ 1]$. We consider the diagonal passing through the origin as the separating surface and assign labels $\{-1, +1\}$ to all the points using it. We changed the labels of 80 points inside the band (width=0.225) around the separating surface.

Fig. 3 shows the reject option classifier learnt using the proposed method. We see that the proposed approach learns the rejection region accurately. We also observe that all of the support vectors are near the two parallel hyperplanes.

Table 2. Behavior of $\boldsymbol{\gamma}'^*$ and $\boldsymbol{\gamma}''^*$

Condition	$\gamma_n'^* \in$	$\gamma_n''^* \in$
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) \in (\rho+\mu, \infty)$	0	0
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) = \rho+\mu$	$(0, \frac{Cd}{\mu})$	0
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) \in [\rho-\mu^2, \rho+\mu)$	$\frac{Cd}{\mu}$	0
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) \in (-\rho+\mu, \rho-\mu^2)$	0	0
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) = -\rho+\mu$	0	$(0, \frac{C(1-d)}{\mu})$
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) \in [-\rho-\mu^2, -\rho+\mu)$	0	$\frac{C(1-d)}{\mu}$
$y_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) \in (-\infty, -\rho-\mu^2)$	0	0

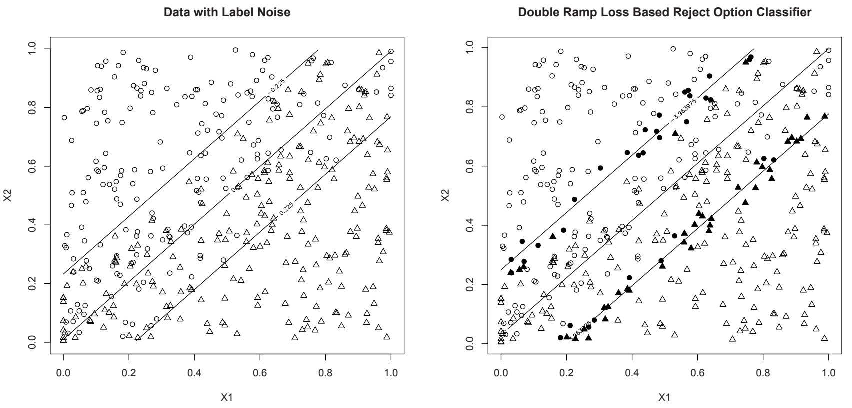


Fig. 3. Left figure shows that label noise affects points near the true classification boundary. Right figure shows reject option classifier learnt using L_{DR} based approach ($C = 100$, $\mu = 1$, $d = .2$). Filled *circles* and *triangles* represent the support vectors.

4 Experimental Results

We show the effectiveness of our approach by showing its performance on several datasets. We also compare our approach with the approach proposed in [7].

4.1 Dataset Description

We report experimental results on 1 synthetic datasets and 2 datasets taken from UCI ML repository [2].

1. **Synthetic Dataset :** Let f_1 and f_2 be two mixture density functions in $\mathbb{R}^2$ defined as follows:

$$f_1(\mathbf{x}) = 0.45\mathcal{U}([1,0]\times[1,1]) + 0.5\mathcal{U}([4,3]\times[0,1]) + 0.05\mathcal{U}([10,0]\times[5,5])$$
$$f_2(\mathbf{x}) = 0.45\mathcal{U}([0,1]\times[1,1]) + 0.5\mathcal{U}([9,10]\times[1,0]) + 0.05\mathcal{U}([0,10]\times[5,5])$$

where $\mathcal{U}(A)$ denotes the uniform density function with support set A. We sample 150 points independently each from f_1 and f_2. We label these points using the hyperplane with $\mathbf{w} = [1 \quad 0]^T$ and $b = 0$. We choose 10% of these points uniformly at random and flip their labels.

2. **Ionosphere Dataset [2] :** This dataset describes the problem of discriminating *good versus bad radars* based on whether they send some useful information about the Ionosphere. There are 34 variables and 351 observations.
3. **Parkinsons Disease Dataset [2] :** This dataset is used to discriminate people with Parkinsons disease from the healthy people. There are 195 feature vectors with each vector having 22 features.

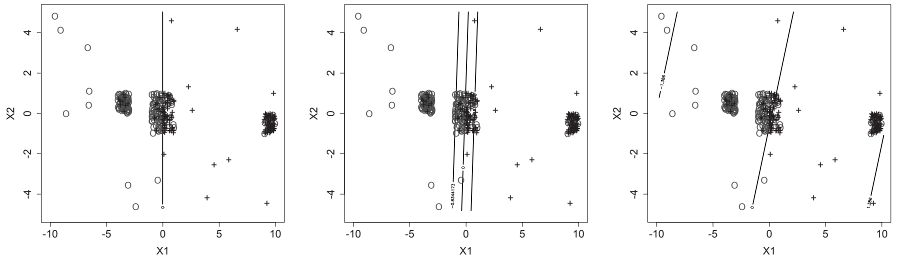
4.2 Experimental Setup

In the proposed $L_{\mathbf{DR}}$ based approach, for solving the dual $D^{(l)}$ at every iteration, we have used the *kernlab* package [9] in **R**. We thank the authors of L_{DH} based method [7] for providing the codes for their approach. For nonlinear problems, we use RBF kernel. In our approach, we set $\mu = 1$. C and σ (width parameter for RBF kernel) are chosen using 10-fold cross validation.

4.3 Simulation Results

We report results for values of d in the interval $[0.05 \quad .5]$ with the step size of 0.05. For every value of d, we find the cross validation risk (under L_{0-d-1}), % accuracy on the non-rejected examples (Acc) and % rejection rate (RR). The results provided are based on 10 repetitions of 10-fold cross validation (CV). We show the average values and standard deviation (computed over the 10 repetitions).

We now discuss the experimental results. Fig. 4(a) shows the Synthetic dataset and the true classification boundary. Fig. 4(b) and (c) show the classifiers learnt using L_{DR} and L_{DH} based approaches respectively for $d = 0.2$. L_{DR} based approach accurately finds the true classification boundary as oppose to



(a) (b) (c)

Fig. 4. (a) Synthetic Dataset and the true classification boundary. Reject option classifiers learnt using (b) proposed L_{DR} based approach for $d = 0.2$, (c) L_{DH} based approach for $d = 0.2$.

Table 3. Comparison results on Synthetic dataset (linear classifiers for both the approaches)

d	$L_{\mathbf{DR}}$ $(C = 2)$			$L_{\mathbf{DH}}$ $(C = 32)$		
	Risk	**RR**	**Acc(unrej)**	**Risk**	**RR**	**Acc(unrej)**
0.05	0.068±0.015	90.87±5.79	75.87±7.95	**0.05**	100	NA
0.1	0.138±0.023	70.35±12.18	79.05±6.87	**0.105**±0.002	95.53±1.69	77.20±6.06
0.15	**0.135**±0.003	65.41±5.06	89.66±0.90	0.136	72.77±0.23	90.56±0.66
0.2	**0.155**±0.006	43.18±4.31	88.56±0.75	0.17	72.67	90.36±1.44
0.25	**0.164**±0.014	32.13±8.43	87.97±1.42	0.204±0.003	66.5±1.7	91±0.74
0.3	**0.148**±0.012	13.23±7.52	87.67±0.69	0.197	46.73±0.14	89.37±0.32
0.35	**0.134**±0.005	4.57±1.80	87.68±0.23	0.21±0.002	43.33±0.65	90.02±0.38
0.4	**0.131**±0.003	1.51±0.56	87.29±0.30	0.21±0.006	31.17±1.26	87.41±0.55
0.45	**0.128**±0.002	0.86±0.45	87.45±0.25	0.265±0.008	9.13±1.1	75.58±0.98
0.5	**0.136**±0.01	0	86.41±0.99	0.297±0.004	0	70.27±0.44

Table 4. Comparison results on Ionosphere dataset (nonlinear classifiers using RBF kernel for both the approaches)

d	$L_{\mathbf{DR}}$ $(C = 2,\ \gamma = 0.125)$			$L_{\mathbf{DH}}$ $(C = 16,\ \gamma = 0.125)$		
	Risk	**RR**	**Acc(unrej)**	**Risk**	**RR**	**Acc(unrej)**
0.05	**0.025**±0.002	34.84±0.92	98.94±0.31	0.029	52.61±0.73	99.47±0.06
0.1	**0.027**±0.003	8.81±0.32	97.99±0.33	0.047±0.002	43.44±0.85	99.46±0.17
0.15	**0.039**±0.003	5.78±0.57	96.81±0.29	0.042±0.003	24.02±1.62	99.3±0.37
0.2	0.044±0.001	3.46±0.51	96.18±0.15	**0.04**±0.002	17.43±0.59	99.42±0.25
0.25	0.047±0.002	1.76±0.41	95.68±0.23	**0.046**±0.001	14.47±0.79	98.9±0.16
0.3	0.052±0.003	0.92±0.46	95.08±0.35	**0.051**±0.003	12.57±0.75	98.56±0.31
0.35	**0.051**±0.003	0.03±0.09	94.88±0.29	0.054±0.002	9.33±0.59	97.72±0.21
0.4	**0.051**±0.002	0	94.95±0.24	0.054±0.003	6.72±0.86	97.09±0.35
0.45	**0.054**±0.002	0	94.64±0.21	0.055±0.003	3.53±0.41	95.97±0.36
0.5	**0.054**±0.001	0	94.62±0.13	0.055±0.005	0	94.55±0.47

L_{DH} based approach. Also, the reject region found by L_{DR} based approach is the most ambiguous region unlike L_{DH} based approach which rejects almost all the points.

Table 3-5 show the experimental results on all the datasets. We observe the following:

1. We see that the proposed L_{DR} based method outperforms L_{DH} based approach in terms of the risk (expectation of L_{0-d-1}). For Synthetic dataset, except for $d = 0.05$ and 0.1, L_{DR} based method has lower CV risk. Similarly, for Ionosphere dataset, except for $d = 0.2, 0.25$ and 0.3, L_{DR} based method has lower CV risk. For Parkinsons dataset, L_{DR} based method has lower CV risk except for $d = 0.35$.
2. We also observe that L_{DR} based method outputs classifiers with significantly lesser rejection rate for all the datasets and for all values of d.

Thus, the proposed L_{DR} based approach outputs classifiers with lesser risk and lesser rejection rate compared to the L_{DH} based approach.

Table 5. Comparison results on Parkinsons Disease dataset (linear classifiers for both the approaches)

d	L_{DR} $(C = 32)$			L_{DH} $(C = 32)$		
	Risk	**RR**	**Acc(unrej)**	**Risk**	**RR**	**Acc(unrej)**
0.05	**0.031**±0.002	43.88±0.80	98.33±0.49	0.043±0.001	86.38±0.92	100
0.1	**0.051**±0.004	41.79±0.77	98.07±1.03	0.061±0.002	53.76±1.64	98.61±0.62
0.15	**0.071**±0.002	40.08±1.21	98.14±0.48	0.086±0.004	39.56±1.13	95.8±0.72
0.2	**0.095**±0.004	37.67±1.04	96.99±0.55	0.125±0.008	29.78±2.06	90.86±1.5
0.25	**0.133**±0.009	20.46±2.79	90.26±1.30	0.142±0.004	22.3±1.95	89.02±0.73
0.3	**0.129**±0.01	4.06±2.06	87.83±1.15	0.131±0.009	14.19±1.05	89.76±1.01
0.35	0.134±0.007	2.49±1.04	87.19±0.76	**0.133**±0.004	9.97±1.18	89.10±0.57
0.4	**0.131**±0.008	0.56±0.44	87.06±0.75	0.133±0.006	6.10±1.62	88.53±0.92
0.45	**0.133**±0.013	0.05±0.17	86.72±1.28	0.14±0.009	2.92±1.09	86.96±1.05
0.5	**0.133**±0.009	0	86.65±0.94	0.139±0.008	0	86.06±0.76

5 Conclusion and Future Work

In this paper, we have proposed a new loss L_{DR} (*double ramp*) for learning the reject option classifier. L_{DR} gives tighter upper bound for L_{0-d-1} compared to convex losses L_{DH} and L_{GH}. Our approach learns the classifier by minimizing the regularized *risk* under the double ramp loss which becomes an instance of DC optimization problem. Our approach can also learn nonlinear classifiers by using appropriate kernel function. Experimentally, we have shown that our approach works superior to L_{DH} based approach for learning reject option classifiers.

References

1. An, L.T.H., Tao, P.D.: Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms. Journal of Global Optimization **11**, 253–285 (1997)
2. Bache, K., Lichman, M.: UCI machine learning repository (2013)
3. Bartlett, P.L., Wegkamp, M.H.: Classification with a reject option using a hinge loss. Journal of Machine Learning Research **9**, 1823–1840 (2008)
4. Chow, C.K.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory **16**(1), 41–46 (1970)
5. Fumera, G., Roli, F.: Support Vector Machines with Embedded Reject Option. In: Lee, S.-W., Verri, A. (eds.) SVM 2002. LNCS, vol. 2388, pp. 68–82. Springer, Heidelberg (2002)
6. Ghosh, A., Manwani, N., Sastry, P.S.: Making risk minimization tolerant to label noise. CoRR, abs/1403.3610 (2014
7. Grandvalet, Y., Rakotomamonjy, A., Keshet, J., Canu, S.: Support vector machines with a reject option. In: NIPS, pp. 537–544 (2008)
8. Herbei, R., Wegkamp, M.H.: Classification with reject option. The Canadian Journal of Statistics **34**(4), 709–721 (2006)
9. Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab - an S4 package for kernel methods in R. Journal of Statistical Software **11**(9), 1–20 (2004)

10. Manwani, N., Sastry, P.S.: Noise tolerance under risk minimization. IEEE Transactions on Systems, Man and Cybernetics: Part-B, 43, 1146–1151 (2013)
11. Ong, C.S., An, L.T.H.: Learning sparse classifiers with difference of convex functions algorithms. Optimization Methods and Software (ahead-of-print), 1–25 (2012)
12. Wegkamp, M., Yuan, M.: Support vector machines with a reject option. Bernaulli **17**(4), 1368–1385 (2011)
13. Yuan, M., Wegkamp, M.: Classification methods with reject option based on convex risk minimization. Journal of Machine Learning Research **11**, 111–130 (2010)